PATENT
Attorney Docket No.: SONY-29000

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:

Edward Eytchison *et al.*

Serial No.: 10/763,866

Filed: January 22, 2004

For: **METHODS AND APPARATUSES FOR AUTOMATIC ADAPTATION OF DIFFERENT PROTOCOLS**

Group Art Unit: 2446

Examiner: Ali, Farhad

RESPONSE TO NOTIFICATION OF NON-COMPLIANT APPEAL BRIEF

162 North Wolfe Road
Sunnyvale, California 94086
(408) 530-9700

Customer No.: 28960

Mail Stop Appeal Brief-Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

Please review the response to the notice of non-compliant appeal brief below.

1. SUMMARY OF CLAIMED SUBJECT MATTER

The invention disclosed in the present application number 10/763,866 is directed to methods and apparatuses for translating commands formatted in different protocols into a common application programming interface. The methods and apparatuses detect at least one device; detect a protocol associated with each device; match the protocol with a protocol translator module; and translate a command formatted in the protocol into a translated command formatted in a common application programming interface through the protocol translator module.

The elements of Claim 1, directed to one embodiment of the presently claimed invention, are described in the Specification at least at page 17, lines 5-21, page 9, line 20 to page 10, line 10, and the accompanying Figures 5-7. The method described there comprises searching for at least one device (110) based on a content type [Present Specification, page 17, lines 5-7], detecting the at least one device (110) [Present Specification, page 17, lines 5-7], detecting a protocol associated with each device (110) [Present Specification, page 17, lines 8-9], matching the detected protocol with a protocol translator module [Present Specification, page 17, lines 11-16], and using the protocol translator module to translate a command formatted in the protocol into a translated command formatted in a common application programming interface [Present Specification, page 17, lines 17-21], wherein the common application programming interface is a single application programming interface that is configured to be used by a plurality of applications (310) [Present Specification, page 9, line 20 to page 10, line 10].

The elements of Claim 7, directed to one embodiment of the presently claimed invention, are described in the Specification at least at page 4, lines 14-21, page 9, line 20 to page 14, line 12, and the accompanying Figures 2-4. The system (300) described there comprises means for searching for at least one device (110) based on a content type [Present Specification, page 14, lines 6-12], means for detecting the at least one device (110) [Present Specification, page 4, lines 14-21], means for detecting a protocol associated with each device (110) [Present Specification, page 13, lines 21-23], means for matching the detected protocol with a protocol translator module [Present Specification, page 12, lines 3-11], and means for using the protocol translator module to translate a command formatted in the protocol into a translated command formatted in a common application programming interface [Present Specification, page 11, line 17 to page 12, line 11], wherein the common application programming interface is a single application programming interface that is configured to be used by a plurality of applications (310) [Present Specification, page 9, line 20 to page 10, line 10].

Means for searching for at least one device (110) based on a content type is shown in Figures 3 and 4. Figure 4 is a simplified block diagram illustrating exemplary services, devices and content organized into classes. In one embodiment, these classes are utilized by the system (300) to encapsulate and categorize information corresponding to unique content, devices, or network services that are easily accessed via two or more different network protocols by an illustrative application using a single, common API. These classes include a device manager class (410), a device class (420), and a service class (430). [Present Specification, page 13, lines 7-13] In one embodiment, the device manager class (410) groups devices in response to a GetDeviceByName command that searches the multiple networks for a specific device. [Present Specification, page 14, lines 6-12]

Means for detecting the at least one device (110) is shown in Figure 3. For example, in one instance the software agent (a "control point" application in UPnP terminology) detects UPnP networks by sending out multicast M-search messages to discover devices. Once an available network is detected, the software agent registers (e.g., in the WINDOWS registry) the available network protocol type and the name of the associated translator module, and then the software agent copies the associated translator module from a central repository to an accessible location in preparation for use. [Present Specification, page 4, lines 14-21]

Means for detecting a protocol associated with each device (110) is shown in Figure 4. In one embodiment, these classes are utilized by the system (300) to encapsulate and categorize information corresponding to unique content, devices, or network services that are easily accessed via two or more different network protocols by an illustrative application using a single, common API. These classes include a device manager class (410), a device class (420), and a service class (430). [Present Specification, page 13, lines 7-13] In one embodiment, the device manager class (430) groups devices in response to a GetDeviceList command that retrieves a list of devices that function using one or more specified network protocols. [Present Specification, page 13, lines 21-23]

Means for matching the detected protocol with a protocol translator module is shown in Figure 3. In other embodiments, the protocol translation layer (350) handles the translation of commands formatted in a plurality of different protocols into translated commands formatted in the common application programming interface. The protocol translation layer (350) supports more than one network protocol. For example, in one instance the protocol translation layer (350) stores more than one translation module for translating commands in multiple different protocols into the common application programming interface. In another instance, the protocol translation layer (350) retrieves an appropriate translation module in response to the protocol to be translated. [Present Specification, page 12, lines 3-11]

Means for using the protocol translator module to translate a command formatted in the protocol into a translated command formatted in a common application programming interface is shown in Figure 3. In some embodiments, the protocol translation layer (350) translates commands utilizing at least one underlying protocol into translated commands utilizing a common application programming interface suitable for use the applications (310), the presentation layer (320), the audio/visual service module (330), and/or the non-audio/visual service module (340). In other embodiments, the protocol translation layer (350) handles the translation of commands formatted in a plurality of different protocols into translated commands formatted in the common application programming interface. [Present Specification, page 11, line 17 to page 12, line 11]

The elements of Claim 8, directed to one embodiment of the presently claimed invention, are described in the Specification at least at page 9, line 20 to page 10, line 10, page 17, lines 5 to 21, and the accompanying Figures 5-7. The method of described there comprises searching for at least one service based on a content type [Present Specification, page 17, lines 5-7], detecting at the least one service [Present Specification, page 17, lines 5-7], detecting a protocol associated with each service [Present Specification, page 17, lines 8-9], matching the detected protocol with a protocol translator module [Present Specification, page 17, lines 11-16], and using the protocol translator module to translate a command formatted in the protocol into a translated command formatted in a common application programming interface [Present Specification, page 17, lines 17-21], wherein the common application programming interface is a single application programming interface that is configured to be used by a plurality of applications (310) [Present Specification, page 9, line 20 to page 10, line 10].

The elements of Claim 9, directed to one embodiment of the presently claimed invention, are described in the Specification at least at page 9, line 20 to page 10, line 10, page 17, lines 5 to 21, and the accompanying Figures 5-7. The method described there comprises searching for a specific device (110) from a plurality of devices (110) based on a content type [Present Specification, page 17, lines 5-7], detecting the plurality of devices (110) wherein each unique device (110) communicates using a corresponding protocol [Present Specification, page 17, lines 5-7], displaying an indication of each device (110) if a protocol translator module is matched with the corresponding protocol [Present Specification, page 17, lines 11-16], and translating a command formatted in the corresponding protocol into a translated command formatted in a common application programming interface through the protocol translator module [Present Specification, page 17, lines 17-21], wherein the common application programming interface is a single application programming interface that is configured to be used by a plurality of applications (310) [Present Specification, page 9, line 20 to page 10, line 10].

The elements of Claim 17, directed to one embodiment of the presently claimed invention, are described in the Specification at least at page 9, line 20 to page 10, line 10, page 19, line 16 to page 20, line 17, and the accompanying Figure 7. The method described there comprises identifying a plurality of protocol translator modules wherein each protocol translator module is associated with a unique protocol [Present Specification, page 19, lines 18-23], storing a list representing the plurality of protocol translator modules [Present Specification, page 20, lines 1-3], displaying an indication of each device (110) having a device protocol that is compatible with one of the plurality of protocol translator modules in the list [Present Specification, page 20, lines 4-9], and translating a command formatted in the device protocol into a translated command formatted in a common application programming interface through one of the plurality of protocol translator modules [Present Specification, page 20, lines 10-12], wherein the common application programming interface is a single application programming interface that is configured to be used by a plurality of applications (310) [Present Specification, page 9, line 20 to page 10, line 10].

The elements of Claim 20, directed to one embodiment of the presently claimed invention, are described in the Specification at least at page 10, line 11 to page 12, line 14, and the accompanying Figures 2–4. The system (300) described there comprises a plurality of applications (310) configured for operating through a single common application programming interface [Present Specification, page 11, lines 17-21], a first device (110) configured for operating using a first protocol, a second device (110) configured for operating using a second protocol [Present Specification, page 10, 11-19] and a protocol translation layer (350) configured for searching for a first protocol translation module corresponding to the first protocol and for searching for a second protocol translation module corresponding to the second protocol [Present Specification, page 11, line 17 to page 12, line 11], the first protocol translation module and second protocol translation module stored in a list representing a plurality of protocol translator modules [Present Specification, page 12, lines 12-14], wherein the protocol translation layer (350) is configured to translate a first command formatted in the first protocol into a command formatted in the single, common application programming interface for use by one of the plurality of applications (310) and to translate a second command formatted in the second protocol into a command formatted in the single common application programming interface for use by another one of the plurality of applications (310) [Present Specification, page 11, line 17 to page 12, line 11].

The elements of Claim 23, directed to one embodiment of the presently claimed invention, are described in the Specification at least at page 9, lines 10-12, page 11, line 17 to page 12, line 2, page 12, lines 3-14, and the accompanying Figures 2-4. The network protocol translation system (300) described there comprises a processor (208, 211) that executes a plurality of run time processes that use only a single application programming interface for network communication [Present Specification, page 9, lines 10-12], wherein the processor (208, 211) enables at least one of the run time processes to communicate via a first network protocol by executing a first translation module that translates between the first network protocol and the single application programming interface and wherein the processor (208, 211) enables the at least one of the run time processes to communicate via a second network protocol, different from the first network protocol, and executing a second translation module that translates between the second network protocol and the application programming interface [Present Specification, page 11, line 17 to page 12, line 2], further wherein the first translation module and second translation module are stored in a list representing a plurality of protocol translator modules [Present Specification, page 12, lines 3-14].

The elements of Claim 24, directed to one embodiment of the presently claimed invention, are described in the Specification at least at page 9, lines 10-12, page 11, line 17 to page 12, line 2, page 12, lines 3-14, and the accompanying Figures 2-4. The method, executed on a computing platform, described there comprises the acts of executing a plurality of run time processes [Present Specification, page 9, lines 10-12] that uses only a single application programming interface for network communication [Present Specification, page 11, line 17 to page 12, line 11], enabling at least one of the run time processes to communicate via a first network protocol by executing a first translation module that translates between the first network protocol and the single application programming interface and enabling the at least one of the run time processes to communicate via a second network protocol, different from the first network protocol, by executing a second translation module that translates between the second network protocol and the single application programming interface [Present Specification, page 11, line 17 to page 12, line 2], wherein the first translation module and second translation module are stored in a list representing a plurality of protocol translator modules [Present Specification, page 12, lines 3-14].

REMARKS

Within the Notification of Non-compliant Appeal Brief, it has been stated that the summary of the claimed subject matter does not contain a concise explanation of the subject matter where each and every limitation is mapped to the Specification. By the above correction, each of the limitations have been individually mapped to the Present Specification. Therefore, the Notification of Non-compliant Appeal Brief should be withdrawn so that the Examiner is able to provide an Answer to the Appeal Brief.

Respectfully submitted,
HAVERSTOCK & OWENS LLP

Dated: March 7, 2010

By: /Jonathan O. Owens/
Jonathan O. Owens
Reg. No.: 37,902
Attorney for Applicant